

グラフ理論

2011.6/13 徳久雅人

1 グラフの基本概念

グラフ理論では、**頂点**と**辺**で構成される図形を**グラフ**という。ただし、辺を**矢印**で表すことや、辺に**標識**を付与することがある。

用語が混乱するので、まとめておこう。以下のとおり複数の言い方がある。

- **頂点, vertex** : 点, 節点, ノード, node
- **辺, エッジ, edge** : 弧, アーク, arc, リンク, link
 - ◇ **無向辺, undirected-edge**
 - ◇ **有向辺, 矢印, directed-edge**
- **標識**: ラベル

全ての辺が有向辺であるグラフを**有向グラフ**という。全ての辺が無向辺であるグラフを**無向グラフ**という。有向辺と無向辺を混在させることは無い。辺に標識が付けられたグラフを**標識付きグラフ**という。

2 集合によるグラフの記述

2.1 基本的な記述

最も基本的なグラフは、**頂点集合 V** と**辺集合 E** であるので、 $G = (V, E)$ と記述する。グラフ G を構成する頂点集合を記述するために、 $V(G)$ と記述する。同様にグラフ G の辺集合を記述するために $E(G)$ と記述する。

辺集合の要素、すなわち、辺 e は、 $e \in V \times V$ である。つまり、 $u, v \in V$ とするとき、 $e = (u, v)$ となる。ここで、頂点 u, v を辺 e の**端点**という。さらに、辺 e が有向辺である場合、 u を**始点**、 v を**終点**という。一方、辺 e が無向辺である場合、 $e = \{u, v\}$ または $e = \{u\}$ と記述することがある。 $\{u\}$ とは始点と終点が同じ場合である。

辺を頂点の 2 つ組で定義した $E \subseteq V \times V$ では、2 つの頂点間に複数の辺が存在することが記述できない。そこで標識が必要になる。標識付きグラフ G は、頂点集合 V 、標識集合 C 、辺集合 E で定義され、 $G = (V, C, E)$ と記述する。ここで、 $E \subseteq V \times V \times C$ である。

例 1

図 1 のグラフを記述しよう。

$$G = (V, C, E)$$

$$V = \{v_1, v_2, v_3, v_4\}$$

$$C = \{e_1, e_2, e_3, e_4, e_5\}$$

$$E = \{(v_1, v_2, e_1), (v_2, v_3, e_3), (v_2, v_4, e_4), (v_4, v_1, e_2), (v_4, v_3, e_5)\}$$

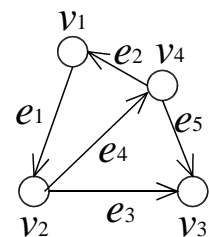


図 1 標識付き有向グラフ

誤解の生じない場合、 (u, v, e_i) を $e_i = (u, v)$ と解釈し、 $G = (V, E)$ で記述することがある。

2.2 辺の端点

辺の端点を記述するために、次の記述方法がある。

- グラフ: $G = (V, E, \partial)$

∂ は、グラフ G における辺 e に対する端点の関数である。 G が無向グラフの場合、 $\partial: E \rightarrow \{X \subseteq V \mid |X| \leq 2\}$ である。 G が有向グラフの場合、 $\partial: E \rightarrow V \times V$ である。

- 有向グラフ: $G = (V, E, \partial^+, \partial^-)$

有向グラフについてはさらに正確な記述がある。 ∂^+ は、有向辺 e に対する始点の関数であり、 $\partial^+: E \rightarrow V$ である。 ∂^- は、有向辺 e に対する終点の関数であり、 $\partial^-: E \rightarrow V$ である。 また、 $\partial(e) = (\partial^+(e), \partial^-(e))$ である。

例 2 図 1 の ∂^+ と ∂^- の関数表をそれぞれ示そう。

$$\partial^+ = \begin{pmatrix} e_1 & e_2 & e_3 & e_4 & e_5 \\ v_1 & v_4 & v_2 & v_2 & v_4 \end{pmatrix} \quad \partial^- = \begin{pmatrix} e_1 & e_2 & e_3 & e_4 & e_5 \\ v_2 & v_1 & v_3 & v_4 & v_3 \end{pmatrix}$$

3 グラフの用語

3.1 頂点と辺に関する用語

- ① **隣接する**: 辺 $e = (u, v)$ のとき、頂点 u と v は隣接している (**adjacent**) という。
- ② **入ると出る**: 辺 $e = (u, v)$ のとき、辺 e は頂点 u から「出る」といい、 v に「入る」という。
- ③ **輪(自己ループ, ループ, loop)**: 辺の端点と同じ頂点であるとき、その辺を輪という。すなわち、 $e = (u, u)$ もしくは、 $e = \{u\}$ 、 $u \in V$ のとき e を輪という。
- ④ **並列辺(多重辺)**: 2 つの辺 e_1, e_2 について、 $\partial^+(e_1) = \partial^+(e_2)$ かつ $\partial^-(e_1) = \partial^-(e_2)$ であることを e_1 と e_2 は並列辺であるという。
- ⑤ **次数**: 頂点 $v \in V$ について、 v に入る辺ののべ数 (**入次数**) と v から出る辺ののべ数 (**出次数**) の合計を次数という。グラフ G において、 v の次数を $\deg_G(v)$ 、出次数を $\deg_G^+(v)$ 、入次数を $\deg_G^-(v)$ と書く。ここで、以下の等式が成り立つ。

$$\deg_G(v) = \deg_G^+(v) + \deg_G^-(v)$$

$$\sum_{v \in V(G)} \deg_G(v) = 2 |E(G)|$$

$$\sum_{v \in V(G)} \deg_G^+(v) = \sum_{v \in V(G)} \deg_G^-(v) = |E(G)|$$

- ⑥ **孤立点**: 次数が 0 の頂点を孤立点という。

3.2 いろいろなグラフ

- ① **単純グラフ**: 輪と並列辺の無いことを**単純**といい、単純なグラフを単純グラフという。
- ② **多重グラフ**: 輪や並列辺のあるグラフを多重グラフという。
- ③ **完全グラフ**: どの 2 頂点間にも辺が存在する単純無向グラフを完全グラフという。頂点が

n 個ある完全グラフを K_n と書く。

- ④ **部分グラフ**: 2つのグラフ G_1, G_2 において, $(V(G_1) \subseteq V(G_2)) \wedge (E(G_1) \subseteq E(G_2))$ であることを $G_1 \subseteq G_2$ と書き, G_1 は G_2 の部分グラフであるという。
- ⑤ **真部分グラフ**: $\{(V(G_1) \subset V(G_2)) \wedge (E(G_1) \subseteq E(G_2))\} \vee \{(V(G_1) = V(G_2)) \wedge (E(G_1) \subset E(G_2))\}$ であることを $G_1 \subset G_2$ と書き, G_1 は G_2 の真部分グラフであるという。
- ⑥ **制限グラフ**: グラフ $G = (V, E)$ および頂点集合 $V' \subseteq V$ に関して, グラフ $G' = (V', E \cap (V' \times V'))$ というグラフ G' を, G の V' への制限グラフといい, $G \upharpoonright V'$ と書く。言い換えると G から V' 以外の頂点を削除し, 削除された頂点に接続する辺を削除して得られるグラフが G' (制限グラフ) である。
- ⑦ **2部グラフ**: グラフ $G = (V, E)$ が2部グラフであるとは, $V_1 \cup V_2 = V, V_1 \cap V_2 = \emptyset, E \subseteq (V_1 \times V_2) \cup (V_2 \times V_1)$ を満たすことである。また, G を V_1 と V_2 を結ぶ2部グラフという。2部グラフ G を $G = (V_1, V_2, E)$ と書く。
- ⑧ **完全2部グラフ**: 2部グラフ (V_1, V_2, E) において, V_1 と V_2 の頂点間全てに辺があるならば完全2部グラフという。 K_{n_1, n_2} と書く。ここで, $n_1 = |V_1|, n_2 = |V_2|$ である。
- ⑨ **グラフの和**: 2つのグラフ $G_1 = (V_1, E_1), G_2 = (V_2, E_2)$ の和 $G_1 \cup G_2$ をグラフの和という。 $G_1 \cup G_2 = (V_1 \cup V_2, E_1 \cup E_2)$ である。
- ⑩ **グラフの積**: 2つのグラフ $G_1 = (V_1, E_1), G_2 = (V_2, E_2)$ の和 $G_1 \cap G_2$ をグラフの積という。 $G_1 \cap G_2 = (V_1 \cap V_2, E_1 \cap E_2)$ である。
- ⑪ **無向基礎グラフ**: 有向グラフ G の全ての辺について, 有向辺を無向辺に置き換えて作成される無向グラフを, G の無向基礎グラフという。

3.3 頂点や辺の並び

グラフ G において, 隣接する頂点とその間の辺の系列 $W = (v_0, e_1, v_1, e_2, v_2, \dots, e_k, v_k)$ (ただし, $(v_{i-1}, v_i, e_i) \in E(G) (i = 1 \dots k)$) について, 次の用語がある。

- ① **ウォーク(walk)**: 上記の W のことをウォーク(walk)という。
- ② **始点と終点**: v_0 を W の始点, v_k を W の終点という。
- ③ **閉じている(closed)**: W の始点と終点と同じ頂点であることを「閉じている(closed)」という。
- ④ **開いている(open)**: W の始点と終点異なることを「開いている(open)」という。

#	W に対する名称	条件
⑤	トレイル(路, 小径, trail)	e_1, e_2, \dots, e_k が全て異なる。
⑥	パス(道, path)	$e_1, e_2, \dots, e_k, v_0, v_1, \dots, v_k$ が全て異なる。
⑦	閉路 (closed trail)	$v_0 = v_k$. e_1, e_2, \dots, e_k が全て異なる。
⑧	閉道 (closed path)	$v_0 = v_k$. $e_1, e_2, \dots, e_k, v_0, v_1, \dots, v_{k-1}$ が全て異なる。
⑨	オイラー路	$E(G)$ の全辺を1回ずつ通過するトレイル。
⑩	オイラー閉路	$E(G)$ の全辺を1回ずつ通過する閉路。
⑪	ハミルトン路	$V(G)$ の全頂点を1回ずつ通過するパス。
⑫	ハミルトン閉路	$V(G)$ の全頂点が v_0, \dots, v_{k-1} に1回ずつ対応する閉道。

かつては、⑥の条件が成立しない場合でも、 W をパスと呼ぶことがあったが、現在では⑥の条件の成立する W であることをパスという。後者のパスを明言するために、**単純パス(simple path)**という呼び方がある。

- ⑬ **長さ(length)**: W を構成する辺の「のべ数」を長さという。1 つの辺の長さを 1 とせず、辺ごとに長さを与えたグラフを用いる場合、各辺の長さの総和を W の長さとする。
- ⑭ **距離(distance)**: 2 つの頂点の間のパスの長さの最小値を距離という。2 つの頂点の間のパスは一般に複数通りあるため、最小値を求める。2 つの頂点が一致するならば、距離は 0 となる。2 つの頂点間にパスが無いならば距離は ∞ となる。

3.4 連結性

- ① **連結**: 無向グラフにおいて、全ての 2 つの頂点の間にパスが存在することを連結という。有向グラフは、その無向基礎グラフが連結であれば、連結であるという。
- ② **強連結**: 有向グラフにおいて、全ての 2 つの頂点の間に有向辺に沿ったパス(有向パス; directed path)の存在することを強連結という。すなわち、頂点 u から頂点 v への有向パスが存在し、かつ、 v から u への有向パスが存在することをいう。

4 行列によるグラフの記述

ここでは、**Incidence Matrix** と **Adjacency Matrix** という 2 種類の行列を紹介する。教科書の訳語が不明確なので、英語で表記する^{*1}。

4.1 Incidence Matrix (接続行列, 入射? 行列)

Incidence matrix とは、行が頂点に対応し、列が辺に対応する行列である。無向グラフのときと有向グラフのときとで定義が異なる。

無向グラフのとき、incidence matrix の i 行 j 列目の要素 r_{ij} は次のように定義する。

$$r_{ij} = \begin{cases} 1 & (\text{頂点 } v_i \text{ が辺 } j \text{ の端点であるとき}) \\ 0 & (\text{頂点 } v_i \text{ が辺 } j \text{ の端点でないとき}) \end{cases}$$

有向グラフのとき、incidence matrix の i 行 j 列目の要素 r_{ij} は次のように定義する。

$$r_{ij} = \begin{cases} 1 & (\text{頂点 } v_i \text{ が辺 } j \text{ の始点であるとき}) \\ -1 & (\text{頂点 } v_i \text{ が辺 } j \text{ の終点であるとき}) \\ 0 & (\text{頂点 } v_i \text{ が辺 } j \text{ の端点でないとき}) \end{cases}$$

4.2 Adjacency Matrix (隣接行列)

Adjacency matrix とは、行と列がどちらも頂点に対応する行列である。

無向グラフのとき、adjacency matrix の i 行 j 列目の要素 s_{ij} は次のように定義する。

$$s_{ij} = \begin{cases} 1 & (\text{頂点 } v_i \text{ と頂点 } v_j \text{ を結ぶ辺が存在するとき}) \\ 0 & (\text{頂点 } v_i \text{ と頂点 } v_j \text{ を結ぶ辺が存在しないとき}) \end{cases}$$

*1 教科書「離散系の数学」での定義(p.46)は古いようだ。

有向グラフのとき, adjacency matrix の i 行 j 列目の要素 s_{ij} は次のように定義する.

$$s_{ij} = \begin{cases} 1 & (\text{頂点 } v_i \text{ から頂点 } v_j \text{ への辺が存在するとき}) \\ 0 & (\text{頂点 } v_i \text{ から頂点 } v_j \text{ への辺が存在しないとき}) \end{cases}$$

例 3

図 1 の incidencey matrix R と adjacency matrix S をそれぞれ示そう.

$$R = \begin{matrix} & \begin{matrix} 1 & 2 & 3 & 4 & 5 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \end{matrix} & \begin{pmatrix} 1 & -1 & 0 & 0 & 0 \\ -1 & 0 & 1 & 1 & 0 \\ 0 & 0 & -1 & 0 & -1 \\ 0 & 1 & 0 & -1 & 1 \end{pmatrix} \end{matrix}$$

$$S = \begin{matrix} & \begin{matrix} 1 & 2 & 3 & 4 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \end{matrix} & \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 \end{pmatrix} \end{matrix}$$

5 グラフの計算や表現

5.1 ウォークの総数を求める

n 個の頂点 v_1, v_2, \dots, v_n をもつグラフ G の adjacency matrix $A = [a_{ij}]$ は, $n \times n$ の行列である. ここで, 行列 A の t 乗を求める.

$$A^t = A \times A \times \dots \times A$$

すると, A^t の i 行 j 列目の要素 $a_{ij}^{(t)}$ は, v_i から v_j への長さ t のウォークの総数となっている.

例 4

例 3 で作成した adjacency matrix S の 2 乗と 3 乗をそれぞれ計算し, ウォークの総数を確認しよう.

$$S^2 = \begin{matrix} & \begin{matrix} 1 & 2 & 3 & 4 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \end{matrix} & \begin{pmatrix} 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} \end{matrix} \qquad S^3 = \begin{matrix} & \begin{matrix} 1 & 2 & 3 & 4 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \end{matrix} & \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{pmatrix} \end{matrix}$$

5.2 行列を2部グラフで表す

n 行 m 列の行列は, 行列の要素 a_{ij} を, 頂点 u_i から頂点 v_j への辺の標識とすることで完全2部グラフ $K_{n,m}$ と同値となる.

例 5

$$A = \begin{matrix} & \begin{matrix} v_1 & v_2 & v_3 & v_4 & v_5 \end{matrix} \\ \begin{matrix} u_1 \\ u_2 \\ u_3 \end{matrix} & \begin{pmatrix} 1 & 4 & 7 & 10 & 13 \\ 2 & 5 & 8 & 11 & 14 \\ 3 & 6 & 9 & 12 & 15 \end{pmatrix} \end{matrix}$$

この行列を2部グラフで表すと図 2 のようになる(標識は3つしか記載していないが...).

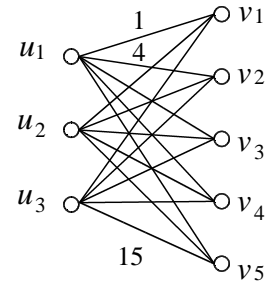


図 2 行列を表す2部グラフ

5.3 最短パスを求める

閉路を含まない単純な有向グラフ $G = (V, E)$ および各辺の長さを表す関数 $\text{length}(e)$ について, 最短距離を求めるアルゴリズムは以下のようになる[1].

```

shortest-path( G, s, t )
  n = | V(G) |      # 頂点の数を n とする
  M = [0..n-1, V]  # 2次元配列 M を準備 (行は 0 から n-1, 列は頂点名のインデクス)
  M[0,t] = 0       # }
  for v ∈ V(G), v ≠ t # } 初期化
    M[0,v] = ∞     # }
  for i = 1, 2, ..., n-1
    for v ∈ V(G)
      M[i,v] = min(M[i-1,v], min(M[i-1,w] + cvw))
                    w ∈ V(G)
    end
  end
  return M[n-1,s]
end

```

ここで, cvw は 2 頂点間 v, w を結ぶ辺の長さを表す, すなわち, $cvw = \text{length}(v,w)$.

頂点数を n , 辺の数を m とする. このアルゴリズムは $O(n^3)$ である. なお, 参考書[2]には, $O(n+m)$ で最短(最長)経路を求めるアルゴリズムが紹介されている.

例 6

図 3 のグラフにおいて、頂点 a から頂点 h までの最短経路を求めよう.

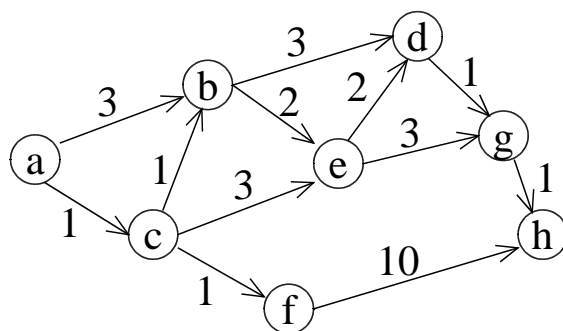


図 3 例題のグラフ

(1) 初期の $M(s = a, t = h, n = 8)$

	a	b	c	d	e	f	g	h
0	∞	∞	∞	∞	∞	∞	∞	0
1								
2								
3								
4								
5								
6								
7								

(2) $i = 1$ の終了後

	a	b	c	d	e	f	g	h
0	∞	∞	∞	∞	∞	∞	∞	0
1	∞	∞	∞	∞	∞	10	1	0
2								
3								
4								
5								
6								
7								

(3) $i = 2$ の終了後

	a	b	c	d	e	f	g	h
0	∞	∞	∞	∞	∞	∞	∞	0
1	∞	∞	∞	∞	∞	10	1	0
2	∞	∞	∞	2	4	10	1	0
3								
4								
5								
6								
7								

(4) $i = 3$ の終了後

	a	b	c	d	e	f	g	h
0	∞	∞	∞	∞	∞	∞	∞	0
1	∞	∞	∞	∞	∞	10	1	0
2	∞	∞	∞	2	4	10	1	0
3	∞	5	7	2	4	10	1	0
4								
5								
6								
7								

(5) $i = 4$ の終了後

	a	b	c	d	e	f	g	h
0	∞	∞	∞	∞	∞	∞	∞	0
1	∞	∞	∞	∞	∞	10	1	0
2	∞	∞	∞	2	4	10	1	0
3	∞	5	7	2	4	10	1	0
4	8	5	6	2	4	10	1	0
5								
6								
7								

(6) $i = 5$ の終了後

	a	b	c	d	e	f	g	h
0	∞	∞	∞	∞	∞	∞	∞	0
1	∞	∞	∞	∞	∞	10	1	0
2	∞	∞	∞	2	4	10	1	0
3	∞	5	7	2	4	10	1	0
4	8	5	6	2	4	10	1	0
5	7	5	6	2	4	10	1	0
6								
7								

(7) $i = 6$ の終了後

	a	b	c	d	e	f	g	h
0	∞	∞	∞	∞	∞	∞	∞	0
1	∞	∞	∞	∞	∞	10	1	0
2	∞	∞	∞	2	4	10	1	0
3	∞	5	7	2	4	10	1	0
4	8	5	6	2	4	10	1	0
5	7	5	6	2	4	10	1	0
6	7	5	6	2	4	10	1	0
7								

(8) $i = 7$ の終了後 ($s=a$, return $M[7, a]$)

	a	b	c	d	e	f	g	h
0	∞	∞	∞	∞	∞	∞	∞	0
1	∞	∞	∞	∞	∞	10	1	0
2	∞	∞	∞	2	4	10	1	0
3	∞	5	7	2	4	10	1	0
4	8	5	6	2	4	10	1	0
5	7	5	6	2	4	10	1	0
6	7	5	6	2	4	10	1	0
7	7	5	6	2	4	10	1	0

5.4 巡回セールスマン問題

最小コストで全ての頂点を通過するウォークを求める問題を、巡回セールスマン問題という。問題設定に幾つかのバリエーションがある。たとえば、始点と終点が指定されていること、始点と終点在同一であること、有向グラフの場合、・・・、などである。

参考文献

- [1] J.Kleinberg, É.Tardos: アルゴリズムデザイン, (訳)浅野孝夫ほか, 共立出版, 2008.
- [2] 浅野孝夫: 離散数学 -グラフ・束・デザイン・離散確率-, 情報学コア・テキスト 2, サイエンス社, 2010.